ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

Tel. (+84.0236) 3736949, Fax. (84-511) 3842771

Website: http://dut.udn.vn/khoacntt, E-mail: cntt@dut.udn.vn

A blue and white logo with black text

Description automatically generated

BÁO CÁO THỰC TẬP TUẦN 3

THỰC TẬP TỐT NGHIỆP

Đề tài:

IPv4/IPv6 Unicast, Multicast, Broadcast, Anycast

|  |  |  |
| --- | --- | --- |
| Họ và tên | Mã số sinh viên | Lớp học phần |
| Thân Nguyên Minh Quân | 102200148 | 20Nh14 |
| Hoàng Huân | 102200131 | 20Nh14 |
| Huỳnh Nguyễn Tấn Sang | 102200230 | 20Nh14 |

CBHD: TS Nguyễn Thế Xuân Ly

Đà Nẵng, 11/2024

MỤC LỤC

[I. **NỘI DUNG TUẦN 3** 3](#_Toc185083488)

[II. **TRIỂN KHAI CÁC YÊU CẦU** 4](#_Toc185083489)

[**2.1. Cài đặt cấu hình môi trường mạng trên Raspberry** 4](#_Toc185083490)

[**2.2. Xây dựng chương trình python gửi nhận thông điệp** 5](#_Toc185083491)

[**2.2.1. IPv4** 5](#_Toc185083492)

[**2.2.2. IPv6** 6](#_Toc185083493)

[**2.2.3. Kết quả** 7](#_Toc185083494)

[**2.2.4. Đánh giá hiệu năng và an toàn** 15](#_Toc185083495)

I. NỘI DUNG TUẦN 3

IPv4/IPv6 Unicast, Multicast, Broadcast, Anycast

1. Xây dựng hệ thống gồm 3 thiết bị Raspberry đã cài đặt sẵn WebOS kết nối qua LAN

2. Xây dựng các chương trình python để gửi/ nhận các thông điệp bằng IPv4 (Unicast, Broadcast và Multicast) //Multicast sẽ sử dụng giao thức IGMP

3. Xây dựng các chương trình python để gửi/ nhận các thông điệp bằng IPv6 (Unicast, Multicast, Anycast)

4. Đánh giá hiệu năng và an toàn để đưa ra sự lựa chọn khi gửi/nhận thông điệp.

II. TRIỂN KHAI CÁC YÊU CẦU

## **2.1. Cài đặt cấu hình môi trường mạng trên Raspberry**

Môi trường gồm 3 RaspberryPi4 trong môi trường mạng Lan: Gồm 1 máy sender và 2 máy receive.   
Cấu hình máy Sender (IP: 192.168.1.101, netmask 255.255.255.0, gateway 192.168.1.1)

Cấu hình 2 máy Receive (IP: 192.168.1.100, netmask 255.255.255.0, gateway 192.168.1.1) và (IP: 192.168.1.102, netmask 255.255.255.0, gateway 192.168.1.1)

Thiết lập multicast cho các máy:

*route add -net 224.0.0.0 netmask 240.0.0.0 dev eth0*

*ifconfig | grep MULTICAST*

*ifconfig eth0 multicast*

*ping 224.0.0.1*

Cấu hình Ipv4 để test Multicast

Máy Sender: 192.168.1.101

Máy Receive 1: 192.168.1.100

Máy Receive 2: 192.168.1.102

MulticastGroup: 224.12.1.1, 224.1.1.1

Cấu hình Ipv6 để test Unicast

Máy Sender: abcd:1234::20/64

Máy Receive 1: abcd:1234::7/64

Máy Receive 2: abcd:1234::8/64

Cấu hình Ipv6 để test Multicast:

Máy Sender: abcd:1234::20/64

Máy Receive 1: abcd:1234::7/64

Máy Receive 2: abcd:1234::8/64

MulticastGroup: ff02:12:1:1, ff02:1:1:1

Cấu hình Ipv6 để test Anycast:

Máy Sender: abcd:1234::20/64

Máy Receive 1: abcd:1234::5/64

Máy Receive 2: abcd:1234::5/64

## **2.2. Xây dựng chương trình python gửi nhận thông điệp**

Sử dụng socket để gửi và nhận gói tin.

### **2.2.1. IPv4**

**- Unicast: Gửi tới địa chỉ đích từ máy nguồn, và chỉ máy đích nhận được.**

# Function Send Unicast(IPv4)

def send\_unicast\_ipv4(message, target\_ip, target\_port):

    with socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM) as s:

        s.sendto(message.encode(), (target7

\_ip, target\_port))

**- Broadcast: Gửi tới toàn mạng.**

# Function Send Broadcast(IPv4)

def send\_broadcast\_ipv4(message, target\_port):

    with socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM) as s:

        s.setsockopt(socket.SOL\_SOCKET, socket.SO\_BROADCAST, 1)

        s.sendto(message.encode(), (255.255.255.255', target\_port))

**- Multicast (Sử dụng giao thức IGMP): Gửi tới nhóm được tham gia.**

**Send: Thay đổi IP nhóm để các thành viên tham gia vào.**

sock=socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM, socket.IPPROTO\_UDP)

sock.setsockopt(socket.IPPROTO\_IP, socket.IP\_MULTICAST\_TTL, MULTICAST\_TTL)

sock.sendto(b"MulticastIPv4forMember",(MCAST\_GRP,MCAST\_PORT))

**Receive: Chỉ định địa chỉ IP nhóm, khi máy send gửi tới thì sẽ nhận được.**

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM, socket.IPPROTO\_UDP)

sock.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

mreq=struct.pack("4sl",socket.inet\_aton(MCAST\_GRP),socket.INADDR\_ANY)

sock.setsockopt(socket.IPPROTO\_IP,socket.IP\_ADD\_MEMBERSHIP, mreq)

### **2.2.2. IPv6**

**- Unicast**

#Function Send Unicast(IPv6)

def send\_unicast\_ipv6(message, target\_ip, target\_port):

    with socket.socket(socket.AF\_INET6, socket.SOCK\_DGRAM) as s:

        s.sendto(message.encode(), (target\_ip, target\_port))

**- Multicast**

**Send:**

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM, socket.IPPROTO\_UDP)

sock.setsockopt(socket.IPPROTO\_IP, socket.IP\_MULTICAST\_TTL, MULTICAST\_TTL)

**Receive:**

mreq = struct.pack("16sI", socket.inet\_pton(socket.AF\_INET6,MCAST\_GRP), 0)

sock.setsockopt(socket.IPPROTO\_IPV6, socket.IPV6\_JOIN\_GROUP, mreq)

**- Anycast**

#Function Send Anycast(IPv6)

def send\_anycast\_ipv6(message, target\_ips, target\_port):

    for ip in target\_ips:

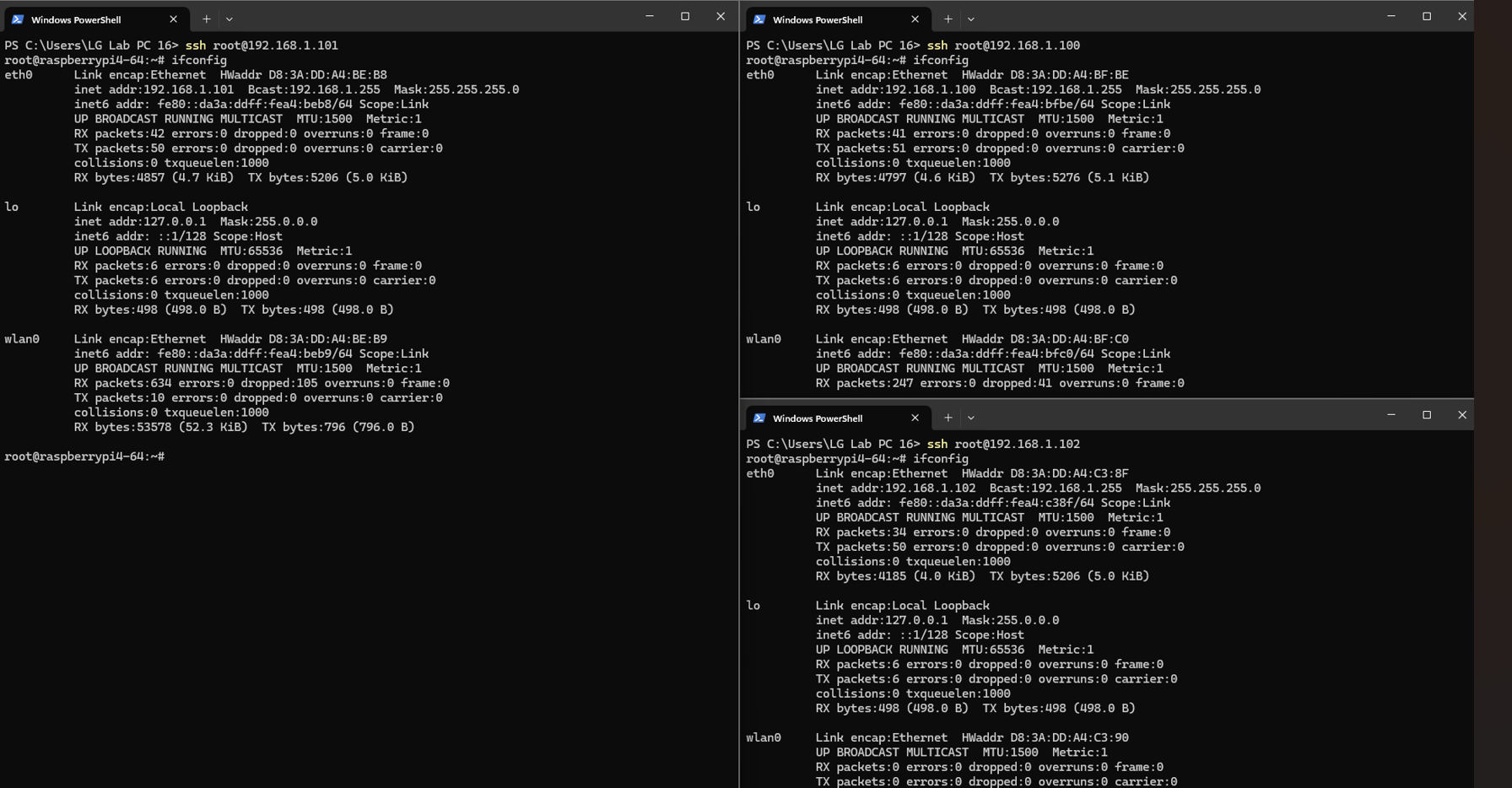
        with socket.socket(socket.AF\_INET6, socket.SOCK\_DGRAM) as s:

            s.sendto(message.encode(), (ip, target\_port))

            print(f"Send Message Anycast to {ip}:{target\_port}")

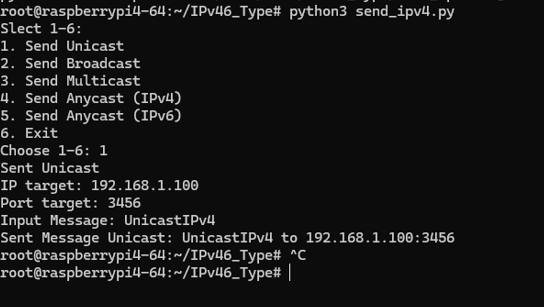
### **2.2.3. Kết quả**

Cấu hình 3 máy trong môi trường mạng:



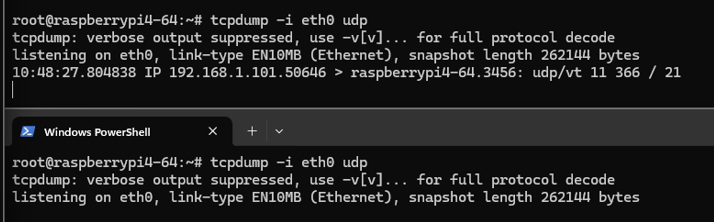
**Unicast (IPv4): là phương thức truyền dữ liệu từ một nguồn duy nhất đến đích duy nhất.**

Chạy code để gửi thông điệp theo Unicast, nhập Iptarget và Port để thông điệp được gửi tới.



Bật tcpdump ở 2 máy còn lại để kiểm tra gói tin có tới địa chỉ 192.168.1.100:3456

không?



Kết quả cho thấy chỉ máy target được nhận thông điệp unicast. Đúng với yêu cầu.

Ngoài ra có thể test nhận và đọc gói tin ở máy target bằng code python.

A computer screen shot of a black screen

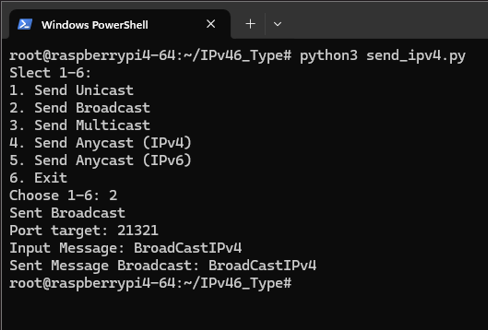
Description automatically generated

Chỉ cần nhập port ở máy target nhận thông điệp.

A black screen with white text

Description automatically generated

**Broadcast (IPv4): là phương thức truyền dữ liệu từ một nguồn đến tất cả các thiết bị trong cùng một mạng.**

****

Gửi gói tin bằng cách chọn port, còn với IP thì set cứng ở trong code cho nó, 192.168.1.255.

A screenshot of a computer

Description automatically generated

Cả hai máy đều nhận được thông điệp gửi đi theo địa chỉ broadcast.

**Multicast (IPv4): là phương thức truyền dữ liệu từ một nguồn đến một nhóm thiết bị được chỉ định. Đây là IP chỉ định của 2 nhóm (224.12.1.1 và 224.1.1.1).**

Code send sẽ gửi thông điệp tới mỗi nhóm và chỉ những ai có trong nhóm thì mới nhận được thông điệp, gửi tới từ máy gửi.

Gửi thông điệp tới nhóm 224.12.1.1 sau đó đổi gửi qua nhóm 224.1.1

A screen shot of a computer program

Description automatically generated

A screenshot of a computer program

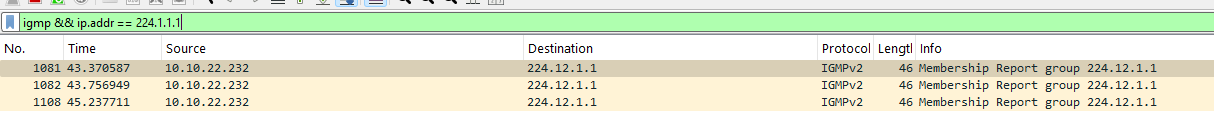
Description automatically generated

Gửi tới nhóm 224.1.1.1 thì chỉ các thành viên có trong nhóm mới nhận được ở đây nhận thông điệp là ‘Multicast IPv4 for Member’, còn khi gửi tới nhóm 224.12.1.1 thì sẽ được nhận thông điệp ‘Multicast IPv4 for Member 224.12.1.1’

**Giao thức IGMP: Dùng trong multicast để quả lí việc tham gia và rời nhóm muticast trong một mạng.**

**- Nhóm 224.12.1.1:**

- Thêm vào nhóm:

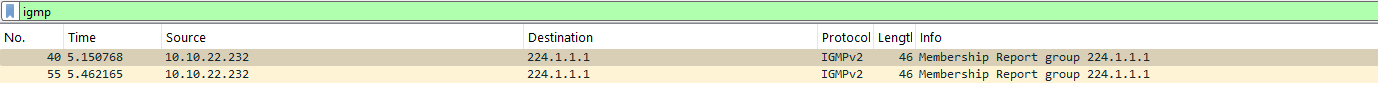


- Rời nhóm: Thông báo qua địa chỉ đích 224.0.0.2 về việc nó rời khỏi nhóm 224.12.1.1

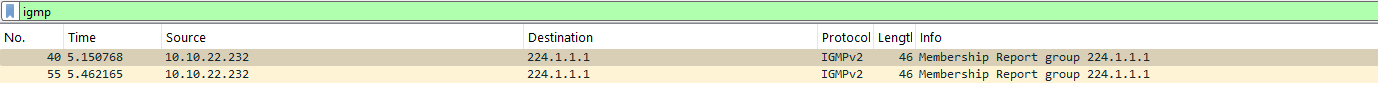


**- Nhóm 224.1.1.1**

Thêm vào nhóm:



Rời nhóm:



**Unicast(IPv6):**

Config IPv6 cho các máy raspberry:

Máy send(abcd:1234::20/64) , hai máy nhận (abcd:1234::7/64 và abcd:2345::5/64) **A screenshot of a computer screen

Description automatically generated**

Gửi thông điệp tới địa chỉ đích. abcd:1234::5:1236

A screen shot of a computer

Description automatically generated

Bắt gói tin bằng tcpdump để kiểm tra

A screen shot of a computer

Description automatically generated

Máy có địa chỉ đích abcd:1234::5 sẽ nhận được thông điệp với ip nguồn của gói tin là abcd:1234::20.

Ngoài ra có thể code Python để nhận và đọc nội dung (nhập port ở máy target)

A screenshot of a computer

Description automatically generated

**Multicast (IPv6):**

Chia ra 2 nhóm để kiểm tra (ff02:12:1:1 và ff02:1:1:1)

Máy gửi sẽ gửi thông điệp tới từng nhóm để kiểm tra.

A screenshot of a computer screen

Description automatically generated

Mỗi nhóm sẽ nhận tin nhắn riêng, chỉ những thành viên thuộc nhóm thì mới nhận được tin nhắn của nhóm đó.

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

**Giao thức ICMP cho IPv6: Hỗ trợ cho các hoạt động quản lí nhóm.**

Nhóm ff02::12:1:1 tham gia và rời:





Nhóm ff02::1:1:1 tham gia và rời:





Thông báo rời nhóm thông qua địa chỉ đích là ff02::2

**Anycast(IPv6): Là phương thức truyền dữ liệu từ một nguồn đến một trong các đích gần nhất( theo đường truyền ...)**

Config 2 máy có cùng địa chỉ IP giống nhau, gửi gói tin, gói tin sẽ được gửi tới máy gần nhất theo nhiều nghĩa. Nhưng chỉ được một máy nhận.

**A screenshot of a computer screen

Description automatically generated**

Gửi gói tin từ máy nguồn: abcd:1234::20/64

A screenshot of a computer

Description automatically generated

Kết quả cho thấy chỉ một máy nhận được thông điệp theo phương thức anycast, và máy nhận được là máy “gần nhất” với máy gửi.

Bây giờ gửi gói tin ở máy ngoài với địa chỉ là abcd:1234::10. Hai IP ở máy nhận vẫn cùng IP abcd:1234::5

A screenshot of a computer

Description automatically generated

Máy gần nhất là máy nhận được gói tin.

### **2.2.4. Đánh giá hiệu năng và an toàn**

Nhóm đã triển khai thành công các loại phương thức gửi ở IPv4 và IPv6, các kết quả cho thấy mọi phương thức đều hoạt động tốt, các thông điệp được gửi nhận thành công, tuy vậy thì sự khác nhau giữa các phương thức gửi về hiệu năng và an toàn là khác nhau, và mỗi loại sẽ phù hợp với môi trường khác nhau.

Đối với môi trường 3 Raspberry trong mạng nội bộ mà nhóm triển khai thì Unicast sẽ là phương thức gửi tốt nhất. Gửi trực tiếp giữa hai, ba thiết bị, hiệu năng tốt nếu có ít thiết bị, nhưng với nhiều thiết bị, băng thông sẽ bị giảm do dữ liệu phải được gửi nhiều lần đến từng thiết bị.

**IPv4:**

Ngoài ra, **Multicast** cũng là phương thức gửi hiệu quả nhất trong các trường hợp có nhiều thiết bị nhận thông tin giống nhau.

Tuy nhiên, **Multicast** yêu cầu cấu hình phức tạp hơn, và khả năng hỗ trợ multicast có thể không có sẵn trên tất cả các mạng hoặc router.

**Broadcast**: Gửi đến tất cả thiết bị trong mạng con. Trường hợp của Raspberry, với phần cứng hạn chế, có thể gặp khó khăn trong việc xử lý quá nhiều broadcast.

Về độ an toàn, **Unicast** nếu không mã hóa và xác thực thì dễ bị tấn công, **Broadcast** là phương thức dễ bị tấn công nhất, do mọi thiết bị trong mạng đều nhận được thông điệp.

**Multicast** là phương thức an toàn, đặc biệt là khi dùng **(IGMP ở IPv4)** để hạn chế truy cập, phát hiện tham gia và rời nhóm bảo vệ dữ liệu khỏi các cuộc tấn công.

**IPv6:**

Với IPv6 thì **Unicast** vẫn là hiệu quả nhất và an toàn nhất lí do là (IPv6 cung cấp các tính năng bảo mật mạnh mẽ hơn so với IPv4 nhờ các cơ chế như **IPsec** (Security Protocol). **Multicast** ở IPv6 sử dụng ICMPv6 để thông báo các hoạt động vào nhóm hoặc rời nhóm, cũng cần các biện pháp để bảo vệ.

**Anycast** có thể giúp tối ưu hóa độ trễ và tăng tính sẵn sàng cho các dịch vụ, vì nó gửi tới nút gần nhất. Khó bị tấn công vì nếu bị DoS chẳng hạn thì gói tin sẽ chuyển qua nút khác gần thứ hai.

**Kết luận:**

Hiệu năng cao nhất: Multicast (IPv4 và IPv6) và Anycast (IPv6) trong môi trường có nhiều thiết bị.

Hiệu quả cao nhất: Unicast trong mạng nhỏ và ít thiết bị, Multicast hoặc Anycast trong môi trường yêu cầu tối ưu hóa độ trễ và khả năng sẵn sàng.